4333004DE USPTO 20003-3529

Translated from the German

Federal Republic of Germany
German Patent Office

Offenlegungsschrift 43 33 004 A1
DE 43 33 004 A1

IPC: **H 04 Q 3/545**
H 04 L 12/00

Date of application: September 28, 1993
Date the 'offenlegungsschrift" was laid open to public inspection: March 30, 1995

Applicant: Siemens AG, D-8033 München, Germany
Inventor: Wolfgang Günther et al.

[Title in German of the object of the invention:]
Weiterleitung von Signalen zwischen Software-Komponenten

ROUTING OF SIGNALS BETWEEN SOFTWARE COMPONENTS

Description

In switching equipments [switching centers], there is required a control according to events [event signals or changes of state], which, e.g., are emitted by subscriber terminals onto the switching equipment. To this end, an event-driven sequence control is used in its capacity as so-called 'embedded system', whereby the hardware of the switching equipment forms the propagation delay-time environment for the sequence control. The sequence control is provided by a freely programmable computer

[calculating machine]. On the one hand, the program of the computer is provided by the user program, and, on the other hand - by the operating system, whereby the operating system forms the interface between the user program and the propagation delay time environment. In the user program, the individual control engineering sequences, such as, e.g., the control of the establishing of the connection or the call [ringing signal] processing, are generated by software components, which are also denoted or referred to as 'controller'. A design language, referred to as 'specification and description language (SDL)', is employed for the generation of the suer program. Software components, which are generated by using the specification and description language (SDL), are denoted as SDL-processes. In the case of a sequence of software components, a routing of signals onto another software component, or an exchange of signals between two software components, is necessary over and over again. The exchange, respectively the forwarding or relay of signals between SDL-processes takes place asynchronously, which is tantamount to having a process, which has routed a signal, controlled up to an end, regardless of an acknowledgement signal.

The invention pertains to a system for the routing of signals between software components in an event-driven sequence control, which is integrated into a propagation delay time ambient environment, and which is generated by a computer (calculating machine), which is freely programmable and is having

a user program and an operating system, whereby the individual control-engineering sequences or runs, such as, e.g., the establishing and disestablishing [clear-down] of connections or the call [ringing signal] processing, are designed as software components, which are also referred to as 'controllers', and whereby the software components are generated by using a design language, and whereby a monitor monitors the propagation delay time ambient surroundings or environment for events or changes of state, and whereby under the terms of a relevant event, a pertinent software component is controlled.

An obvious solution for the purpose of asynchornous routing of a signal from a first software component to a second software component consists in that the signal is controlled by means of the operating system. In doing so, it is becoming disadvantageously noticeable that, on the one hand, for each signal, which is to be routed, a time-consuming process change is required from the user program to the operating system, and from the operating system to the user program, and, on the other hand, for the purposes of SDL-processes of the same kind, which are to be executed simultaneously (such as, e.g., call processing) a copy of these SDL-processes of common software component, which copy is referred to as "instant", is required, which necessitates a correspondingly large input for the provision and management of storage space [main memory location] in the operating system.

The objective to design in such a way the routing of signals

between software components that the cited disadvantages are avoided, forms the basis of the invention.

The set objective is achieved as a result of the fact

- that a first software component writes into a waiting queue [waiting list], belonging to the user program,a signal, which is to be routed to a second software component, together with the designation or identity of the second software component.

- that with the concluding or terminal embodiment of the first software component in the monitor, a return or 'jump back' to the monitor takes place,

- that the monitor interrogates or scans the waiting queue
for the oldest entry, and invokes the signal, stored therein,

- that the signal, which has been invoked, is routed to the second software component, and additionally processed therein,

- that the entry, which has been invoked in the waiting queue, is deleted.

Apart from the preservation of the semantics of the design language, the object of the application - when signals are further processed - produces as a consequence the advantage that only a single operating-system process is opposed or compared to entire control-engineering sequences of the same type. In the case of a sequence control, which makes use of the object of the application, the propagation delay time of a signal between

software components is shortened, as a result of which the dynamic propagation time effectiveness is improved on the whole. Besides this, the asynchronism [asynchrony] is preserved when signals are routed.

By means of a figure, the invention is now described in greater detail in a scope, which is necessary for its understanding.

Fig. 1 is a diagrammatic representation of software modules, which are typically used, and the temporal succession of sequences or runs, occurring therein.

The software modules would be part of a sequence control integrated into an event-driven propagation time delay environment (embedded system). In doing so, the propagation delay time would be provided by means of a switching equipment, which would establish the connections between various subscriber terminals. The events or changes of state might be initiated by subscriber terminals, which, e.g., are in the state of establishing or setting up the connection. From Fig. 1, there can be deduced a first software component SCO1 and a second software component SCO2, which are assigned to the user program. A software program is generated with the help of a procedure, triggered by events of the same kind, to which procedure event-individual local data are assigned, respectively. The procedures themselves are generated by using a design language, in particular 'the specification and description language'. Over

the course of the run of a procedure, the routing of signals to another procedure may be necessary. In accordance with the invention, a signal, which is to be routed from a first software component to a second software component, is routed by means of a waiting queue, MQ (for message queue), which is to be provided in the user program. In detail, the routing of signals from a first software component to a second software component proceeds as follows. As a result of an event or change of state at instant (time) T0, a monitor MO is activated. According to the type of event, at instant T1, the monitor invokes or calls in the corresponding software component (in the example [it is]: SCO1). At instant T2, one ought to proceed inside the software component SCO1, starting from the initial state (IDLE), according to the special event (in the example [it is]: SETUP_IND), with the corresponding branch or splitter. At instant T3, a signal (in the example [it is]: SETUP_ACK_REQ) should be routed to the software component SCO2. To this end, this signal, (SETUP_ACK_REQ), together with the designation or identification of the software component, which should receive the signal (in this case it is the software component (SCO3), is entered into the waiting queue. At instant T4, a signal (SETUP_REQ) should be routed to the software component SCO3, which is not diagrammatically represented in greater detail. To this end, the said signal together with the designation or identification of the software component, which should receive the signal (in this case software

component SCO3) is entered into the waiting queue. At instant T5, the end of the branch in the software component SCO1 is attained, as a result of which it changes over into its new state (in the example [it is]: WAIT_FOR_ACM), and it is returned or 'jumped back' by it to the monitor. At instant T6, the queue - starting with the oldest entry- is read out. The signals, temporarily stored in the queue, are fed to the relevant software components (in the example [those are]: the software component SCO2, respectively the software component SCO2, which is not diagrammatically represented). At time T7, one has to proceed in the software component SCO2 with the corresponding branch or splitter, starting from the actual state, according to the received signal. On the of the branch, the software component SCO2 changes over into a new state (in the example [it is]: CALL PROCRESS), whereby a return to the monitor takes place.

In this case, the invention was solely represented in the example of a switching equipment [exchange equipment] but it is not limited to it. The invention can be sued in each and any kind of event-driven process controls.

CLAIMS

1. System for the routing of signals, between software components (SC) in an event-driven sequential control, which is

integrated into a propagation delay time environment, and which is provided by a freely programmable computer [calculating machine], having a user program and an operating system, whereby the individual control-engineering runs of the user program are designed as software components, generated as a result of using a design language, and whereby a monitor (MO) monitors the propagation delay-time ambient surroundings [environment] for events or changes of state, and whereby a pertinent software component is controlled according to a relevant event,

characterized in that

- a first software component (SCO1) writes [loads] a signal, which is to be routed to a second software component (SCO2), together with the designation or identification of the second software component, into a queue (MQ), belonging to the user program,

- with the concluding embodiment of the first software component, a return ['jump back'] into the monitor takes place,

- the monitor interrogates the queue on the oldest entry, and invokes the signal, stored therein,

- the invoked signal is routed to the second software component, and additionally processed therein,

- the entry, invoked in the queue, is deleted.

2. System, as claimed in claim 1, characterized in that the propagation delay time, is produced or provided by a switching equipment [exchange equipment].

3. System as claimed in claim 1 or 2, characterized in that the design language is provided by the specification and description language.

1 page of drawings

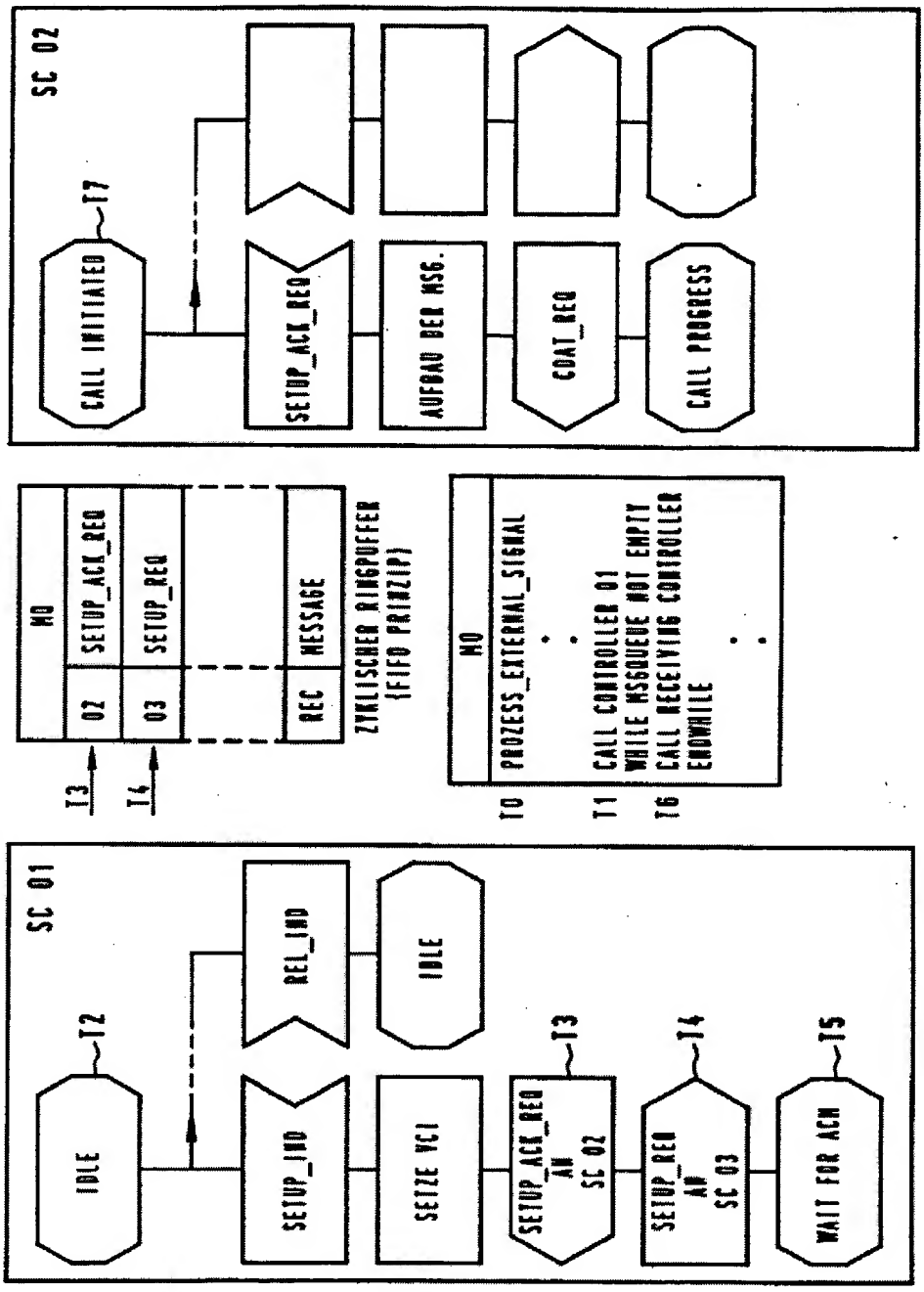
Translator's note: The text and acronyms in the drawings are in English with two exceptions, namely:

The text in German at the very center of the page, "ZYKLISCHER RINGPUFFER (FIFO PRINZIP)", means "cyclic ring buffer store (FIFO Principle);

In the second column [from right to left], the text in third box from top to bottom, namely "AUFBAU der MSG" means 'ESTABLISHING (SETTING UP) of the MSG.]

USDoC/USPTO/STIC/Translations Branch
Translated by John M Koytcheff, MSc (Civ. Engrg. & Water Engrg.)
WHO Fellow (Env. Engrg.), Gr. USNWC/USNC&C
USPTO Translator (GERMAN & Germanic languages)
May 21, 2003

Nummer: DE 43 33 004 A1
Int. Cl.[6]: H 04 Q 3/545
Offenlegungstag: 30. März 1995



508 013/431